SERVLETS - FILE UPLOADING

A Servlet can be used with an HTML form tag to allow users to upload files to the server. An uploaded file could be a text file or image file or any document.

Creating a File Upload Form:

The following HTM code below creates an uploader form. Following are the important points to be noted down:

- The form **method** attribute should be set to **POST** method and GET method can not be used.
- The form **enctype** attribute should be set to **multipart/form-data**.
- The form **action** attribute should be set to a servlet file which would handle file uploading at backend server. Following example is using **UploadServlet** servlet to upload file.
- To upload a single file you should use a single <input .../> tag with attribute type="file". To allow multiple files uploading, include more than one input tags with different values for the name attribute. The browser associates a Browse button with each of them.

```
<html>
<head>
<title>File Uploading Form</title>
</head>
<body>
<h3>File Upload:</h3>
Select a file to upload: <br />
<form action="UploadServlet" method="post"
                     enctype="multipart/form-data">
<input type="file" name="file" size="50" />
<br />
<input type="submit" value="Upload File" />
</form>
</body>
</html>
```

This will display following result which would allow to select a file from local PC and when user would click at "Upload File", form would be submitted along with the selected file:

```
File Upload:
Select a file to upload:




NOTE: This is just dummy form and would not work.
```

Writing Backend Servlet:

Following is the servlet **UploadServlet** which would take care of accepting uploaded file and to store it in directory <Tomcat-installation-directory>/webapps/data. This directory name could also be added using an external configuration

such as a **context-param** element in web.xml as follows:

```
<web-app>
....
<context-param>
    <description>Location to store uploaded file</description>
    <param-name>file-upload</param-name>
    <param-value>
        c:\apache-tomcat-5.5.29\webapps\data\
     </param-value>
</context-param>
....
</web-app>
```

Following is the source code for UploadServlet which can handle multiple file uploading at a time. Before procedding you have make sure the followings:

- Following example depends on FileUpload, so make sure you have the latest version of **commons-fileupload.x.x.jar** file in your classpath. You can download it from http://commons.apache.org/fileupload/.
- FileUpload depends on Commons IO, so make sure you have the latest version of **commons-io-x.x.jar** file in your classpath. You can download it from http://commons.apache.org/io/.
- While testing following example, you should upload a file which has less size than *maxFileSize* otherwise file would not be uploaded.
- Make sure you have created directories c:\temp and c:\apache-tomcat-5.5.29\webapps\data well in advance.

```
// Import required java libraries
import java.io.*;
import java.util.*;

import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.commons.fileupload.FileItem;
import org.apache.commons.fileupload.FileUploadException;
import org.apache.commons.fileupload.disk.DiskFileItemFactory;
import org.apache.commons.fileupload.servlet.ServletFileUpload;
import org.apache.commons.io.output.*;

public class UploadServlet extends HttpServlet {

   private boolean isMultipart;
   private String filePath;
   private int maxFileSize = 50 * 1024;
   private int maxMemSize = 4 * 1024;
   private File file ;

   public void init( ){
      // Get the file location where it would be stored.
      filePath =
             getServletContext().getInitParameter("file-upload");
   }
   public void doPost(HttpServletRequest request,
               HttpServletResponse response)
              throws ServletException, java.io.IOException {
      // Check that we have a file upload request
      isMultipart = ServletFileUpload.isMultipartContent(request);
      response.setContentType("text/html");
      java.io.PrintWriter out = response.getWriter( );
      if( !isMultipart ){
         out.println("<html>");
```

```
         out.println("<head>");
         out.println("<title>Servlet upload</title>");
         out.println("</head>");
         out.println("<body>");
         out.println("<p>No file uploaded</p>");
         out.println("</body>");
         out.println("</html>");
         return;
      }
      DiskFileItemFactory factory = new DiskFileItemFactory();
      // maximum size that will be stored in memory
      factory.setSizeThreshold(maxMemSize);
      // Location to save data that is larger than maxMemSize.
      factory.setRepository(new File("c:\\temp"));

      // Create a new file upload handler
      ServletFileUpload upload = new ServletFileUpload(factory);
      // maximum file size to be uploaded.
      upload.setSizeMax( maxFileSize );

      try{
      // Parse the request to get file items.
      List fileItems = upload.parseRequest(request);

      // Process the uploaded file items
      Iterator i = fileItems.iterator();

      out.println("<html>");
      out.println("<head>");
      out.println("<title>Servlet upload</title>");
      out.println("</head>");
      out.println("<body>");
      while ( i.hasNext () )
      {
         FileItem fi = (FileItem)i.next();
         if ( !fi.isFormField () )
         {
            // Get the uploaded file parameters
            String fieldName = fi.getFieldName();
            String fileName = fi.getName();
            String contentType = fi.getContentType();
            boolean isInMemory = fi.isInMemory();
            long sizeInBytes = fi.getSize();
            // Write the file
            if( fileName.lastIndexOf("\\") >= 0 ){
               file = new File( filePath +
               fileName.substring( fileName.lastIndexOf("\\"))) ;
            }else{
               file = new File( filePath +
               fileName.substring(fileName.lastIndexOf("\\")+1)) ;
            }
            fi.write( file ) ;
            out.println("Uploaded Filename: " + fileName + "<br>");
         }
      }
      out.println("</body>");
      out.println("</html>");
   }catch(Exception ex) {
       System.out.println(ex);
   }
   }
   public void doGet(HttpServletRequest request,
                     HttpServletResponse response)
         throws ServletException, java.io.IOException {

         throw new ServletException("GET method used with " +
                 getClass( ).getName( )+": POST method required.");
   }
}
```

Compile and Running Servlet:

Compile above servlet UploadServlet and create required entry in web.xml file as follows.

```
<servlet>
   <servlet-name>UploadServlet</servlet-name>
   <servlet-class>UploadServlet</servlet-class>
</servlet>

<servlet-mapping>
   <servlet-name>UploadServlet</servlet-name>
   <url-pattern>/UploadServlet</url-pattern>
</servlet-mapping>
```

Now try to upload files using the HTML form which you created above. When you would try http://localhost:8080/UploadFile.htm, it would display following result which would help you uploading any file from your local machine.

```
File Upload:
Select a file to upload:
```

If your servelt script works fine, your file should be uploaded in c:\apache-tomcat-5.5.29\webapps\data\ directory.